Promise Tryout

Problem Statement

* We can see how a new promise is created
* It also shows how to resolve and reject a promise
* Below is the Javascript program where a function is called which returns a promise.Below is the output expected:

**"Let's take a trip to Paris"**

* Try to change the input from "Paris" to "Florida" and observe the output.

function getTrip(location) {

return new Promise(function (resolve, reject) {

if (location == "Paris") {

resolve("Let's take a trip to " + location);

} else {

reject(Error("Invalid Location"));

}

});

}

getTrip("Paris").then(

function (data) {

console.log(data);

},

function (error) {

console.log(error.message);

}

)

Promise Chaining-Demo

Problem Statement

Problem Statement:

* The given code highlights the creation of a Promise Chaining.
* Usage of resolve and reject
* A JavaScript program to book a hotel only after booking a flight.

To achieve this, the promise returned from the bookHotel function is resolved only after resolving the promise from bookFlight function.

If the promise gets rejected from bookflight then it won't execute the second function.

* Change the flight name from "AirIndia" to "Airways" and check the output.

function bookFlight(airline) {

return new Promise(function (resolve, reject) {

if (airline == "AirIndia") {

setTimeout(resolve(5600), 2000);

} else {

reject(Error("Flight can not be booked"))

}

})

}

function bookHotel(flightPrice) {

return new Promise(function (resolve) {

setTimeout(resolve(7000 + flightPrice), 1000);

})

}

bookFlight("AirIndia")

.then(function (flightData) { return bookHotel(flightData) })

.then(function (cumulativeData) { console.log(" Total is " + cumulativeData) })

.catch(e => console.log(e.message))

Promise async/await -Tryout

Problem Statement

To simplify the given code we can use async and await in Promise Chaining.

Copy and replace the getTotal function with given code.

async function getTotal(){

var flightData=await bookFlight();

var cumulativeData=await bookHotel(flightData);

console.log(" Total is " + cumulativeData)

}

function bookFlight() {

return new Promise(function (resolve) {

setTimeout(resolve(5600), 2000);

})

}

function bookHotel(flightPrice) {

return new Promise(function (resolve) {

setTimeout(resolve(7000 + flightPrice), 1000);

})

}

function getTotal(){

bookFlight()

.then(function (flightData) { return bookHotel(flightData) })

.then(function (cumulativeData) { console.log(" Total is " + cumulativeData) })

}

getTotal()